CLAIMS

WHAT IS CLAIMED IS:

1. A client-server computer system comprising:

at least one client;

an application properties server accessible by at least one of said at least one client, said application properties server coupled to said at least one client for providing configuration data in response to a request from said at least one client;

an administration system coupled to said application properties server; and

a storage medium coupled to said application properties server and said administration system for storing said configuration data.

2. A client-server computer system comprising:

a plurality of client application servers operating multiple computer network protocols;

an application server network accessible by said plurality of client application servers via at least one application software protocol, wherein said application server provides configuration information in response to at least one configuration request from at least one client application server; and,

a storage medium coupled to said application server network, said storage medium for storing system current configuration information, wherein said storage medium is administered by a server coupled to said client-server system.

3. A client-server computer system according to claim 2, wherein said storage

medium comprises a database.

4. A client-server computer system according to claim 2, wherein said configuration information is represented by a storage schema in the form of Lightweight Directory Access Protocol.

5. A client-server computer system according to claim 3, wherein said database contains a table-based system of configuration information, wherein said tables are searchable by said application server network in response to a request from at least one client.

6. A client-server computer system according to claim 4, wherein the storage schema represented by Lightweight Directory Access Protocol represents a table-based system configuration information.

7. A client-server computer system according to claim 2, wherein said database stores configuration information that is dynamically updateable by an external administrator.

8. A client-server computer system according to claim 2, wherein said storage schema is in the form of Lightweight Directory Access Protocol and represents configuration information that is dynamically updateable by an external administrator.

9. A client-server computer system according to claim 7, wherein at least one client is coupled to said application server network via an RMI interface.

10. A client-server computer system according to claim 8, wherein at least one client is coupled to said application server network via an RMI interface.

11. A client-server computer system according to claim 3, wherein said configuration information is stored and retrieved from said storage medium via Key Value Pairs.

12. A client-server computer system according to claim 4, wherein said configuration information is stored and retrieved from said storage medium via Hashtable Hierarchy.

13. A client-server computer system according to claim 7, wherein said configuration information is stored and retrieved from said storage medium via Key Value Pairs.

14. A client-server computer system according to claim 8, wherein said configuration information is stored and retrieved from said storage medium via Hashtable Hierarchy.

15. An application server comprising:

a plurality of client application servers operating a plurality of computer protocols;

means for performing configuration services in response to configuration requests from said plurality of client application servers, said means for performing configuration services being coupled to said plurality of client application servers;

means for storing and maintaining a system of configuration information coupled to said means for performing configuration services; and

means for interfacing said plurality of client application servers to said means for performing configuration services.

16. An application server according to claim 15, wherein said means for interfacing said plurality of client application servers to said means for performing configuration services includes a CORBA server application.

17. An application server according to claim 16, wherein said means for interfacing said plurality of client application servers to said means for performing configuration services includes a properties server application for handling RMI requests for configuration services.

18. An application server according to claim 17, wherein said means for interfacing said plurality of client application servers to said means for performing configuration services includes a common database access library.

19. An application server according to claim 18, wherein said means for interfacing said plurality of client application servers to said means for performing configuration services includes a database server coupled to said properties server application for handling RMI requests and said CORBA server application for interacting with said means for storing and maintaining configuration information.

20. An application server according to claim 15, further comprising a Java RMI Application Programming Interface.

21. An application server according to claim 20, further comprising a CORBA gateway.

22. An application server according to claim 21, wherein said means for performing configuration service is implemented by a base Java RMI service in a service broker framework.

23. An application server according to claim 22, wherein said service broker framework is implemented using at least one XML service broker configuration file.

24. An application server according to claim 23, wherein said configuration information is stored and retrieved from said means for storing via Key Value Pairs.

25. An application server according to claim 23, wherein said configuration information is stored and retrieved from said means for storing via Hashtable Hierarchy.

26. A system for providing an application service, the system comprising:

an application server;

at least one application running a Java application and networked with said application server;

at least one application running a CORBA application and networked with said application server;

at least one application running an Internet application and networked with said application server;

one or more application programming interfaces, the one or more application programming interfaces coupled to the application server for receiving data configuration requests via a plurality of computer network protocols; and

at least one dynamically-maintainable data schema coupled to said application server.

27. A client-server computer system according to claim 26, wherein said data schema comprises configuration information and is at least partially in the form of a database.

28. A client-server computer system according to claim 26, wherein said data schema comprises configuration information in the form of Lightweight Directory Access Protocol.

29. A client-server computer system according to claim 27, wherein said application server and said data schema are remotely located to said plurality of client application servers and said configuration information is maintainable by a remote administrator.

30. A client-server computer system according to claim 28, wherein said application server and said data schema are remotely located to said plurality of client application servers and said configuration information is maintainable by a remote administrator.

31. The client-server computer system of claim 29, wherein data passes between said application servers and said application server in the form of a string.

32. The client-server computer system of claim 30, wherein data passes between said application servers and said application server in the form of a string.

33. The client-server computer system of claim 29, wherein data passes between said application servers and said application server in the form of a hashtables.

34. The client-server computer system of claim 30, wherein data passes between said application servers and said application server in the form of a hashtables.

35. A server system for providing data configuration service in response to requests from applications coupled to the server, the system comprising:

a CORBA application server;

an RMI application server;

an internet application server,

wherein said application servers are all in communication with a centralized properties server, the application servers providing configuration requests to the properties server via a network;

one or more application programming interfaces, the one or more application programming interfaces capable of handling a plurality of software protocols in communication with the properties server and said application servers; and

a data schema in communication with said network, for storing configuration variables and accessible by said properties server.

36. A server system according to claim 35, wherein said data schema comprises an ORACLE database.

37. A server system according to claim 35, wherein said configuration variables

comply with Lightweight Directory Access Protocol.

38. A server system according to claim 35, wherein said one or more interfaces includes a CORBA server application.

39. A server system according to claim 38, wherein said one or more interfaces includes a server application for handling RMI requests.

40. A server system according to claim 39, wherein said one or more interfaces includes a common database access library.

41. A server system according to claim 40, wherein said one or more interfaces includes a database server coupled to said properties server.

42. A server system according to claim 35, further comprising a Java RMI Application Programming Interface.

43. A server system according to claim 42, further comprising a CORBA gateway.

44. An server system according to claim 43, wherein said properties server is implemented by a base Java RMI service in a service broker framework.

45. A server system according to claim 44, wherein said service broker framework is implemented using at least one XML service broker configuration file.

46. A system for providing an application service, the system comprising:

means for receiving a service request from a customer, the customer requesting configuration information from said system;

means for sending a configuration request instruction to an application server corresponding to the configuration information;

means for sending a service request from the application server to a relational database, the service response based at least in part on the configuration request;

means for performing configuration services in response to said service request;

means for updating said relational database based on current configuration requirements of said system;

means for sending a configuration result from the application server to said customer based at least in part on the configuration request; and

means for providing a response to said system from said customer to said system in response to said configuration result.

47. A computer-readable medium storing a plurality of instructions adapted to be executed by a processor for providing an application service, the plurality of instructions comprising instructions to:

receive a service request from a customer data device, the customer data device including an instruction concerning configuration files;

generate a service session instruction, the service session instruction based at least in part on the service request;

send the service session instruction to one or more open application programming interfaces, the service session instruction corresponding to one or more data configuration requests from said customer data device;

perform one or more configuration functions based on stored variables in a relational database; and

send a configuration service response to the customer data device, the configuration service response based on the service request.

48. A medium according to claim 47, wherein said relational database comprises an Oracle database and further comprises an instruction to load at least a portion of said database into a memory upon startup of said application service.

49. A medium according to claim 47, wherein said variables are stored in the format of Lightweight Directory Access Protocol and further comprise an instruction to load variables into a memory upon startup of said application service.

50. A method of providing a configuration service with a client-server computer system comprising the steps of:

coupling a configuration request between a client application server and an application server;

providing a configuration service request instructions to a data schema in response to said configuration request coupled between said client application server and said application server;

retrieving updated configuration files from a storage mass coupled to said application server; and

coupling a response to said client application server.

51. A method for providing an application service, the method comprising:

a step for sending a data configuration service request from a user;

a step for generating a configuration service instruction, the service instruction based at least in part on a configuration service request from said user;

a step for sending said configuration service instruction to one or more data schemas via one or more application programming interfaces, the service instruction corresponding to one or more configuration requests from the user;

a step for dynamically updating a table of configuration files stored in said one or more data schemas based on changes to the application service;

a step for calling up at least one table of configuration files from said one or more data schemas;

a step for providing configuration information to said user in accordance with updated configuration files stored in said one or more data schemas; and

a step for sending service response to the user requesting configuration service.

52. A method according to claim 51, further comprising the step of initializing at least one property server object.

53. A method according to claim 52, further comprising the step of creating a connection between said at least one property server object and at least one of said data schemas.

54. A method according to claim 53, further comprising the step of pooling at least one server object with at least one client application server.

55. A method according to claim 51, further comprising the steps of establishing a service broker framework by initializing a plurality of configurable property server objects and pooling said objects to a plurality of client application servers.

56. A method according to claim 55, further comprising the steps of reading an XML configuration file and establishing a service for at least one service tag in said XML configuration file.

57. A method according to claim 51, further comprising the step of handling configuration files using Key Value Pairs.

58. A method according to claim 51, further comprising the step of handling configuration files using Hashtable Hierarchy.

59. A method according to claim 51, further comprising retrieving configuration files through Java Database Connectivity API.

60. A method for providing an application service, the method comprising:

a step for sending a data configuration service request from a user;

a step for generating a configuration service instruction, the service instruction based at least in part on a configuration service request from said user;

a step for sending the service instruction to one or more data storage schemas via one or more application programming interfaces, the service instruction corresponding to one or more configuration requests from the user;

a step for dynamically updating a table of configuration files stored in said one or more data schemas based on changes to the application service;

a step for calling up at least one table of configuration files from said one or more data storage;

a step for providing configuration information to said user in accordance with updated configuration files stored in said storage schemas; and

a step for sending service response to the user requesting configuration service.

61. A method according to claim 60, further comprising the steps of:

establishing a service broker framework;

initializing a plurality of configurable property server objects; and

pooling said plurality of configurable property server objects to at least one client application server.

62. A method according to claim 61, further comprising the steps of:

reading an XML configuration file; and

establishing a service for at least one service tag in said XML configuration file.